Closures

Closures

A function within a function

Closures

A function within a function

An anonymous function, lambda expression

```
|a: i32, b: i32| println!(“{}”, a + b);
|a: i32, b: i32| -> i32 { a + b };
```

Closures

A function within a function

An anonymous function, lambda expression

```
|a: i32, b: i32| println!(“{}”, a + b);
|a: i32, b: i32| -> i32 { a + b };
```

A function can be assigned to a variable

```
let sum = |a: i32, b: i32| -> i32 {a + b};
sum(2, 3);
```

Closures

A function within a function

An anonymous function, lambda expression

```
|a: i32, b: i32| println!(“{}”, a + b);
|a: i32, b: i32| -> i32 { a + b };
```

A function can be assigned to a variable

```
let sum = |a: i32, b: i32| -> i32 a + b;
sum(2, 3);
```

A clojure can be generic

```
let gen = |x| { println!("received {}", x) };
gen(3);
```